

Digital Object Identifier

Exploring the U-Net++ model for Automatic Brain Tumor Segmentation

NEIL MICALLEF¹, (Graduate Student Member, IEEE), DYLAN SEYCHELL², (Senior Member, IEEE) and CLAUDE J. BAJADA³

¹Department of Artificial Intelligence, University of Malta, Msida, Malta (e-mail: neil.micallef.14@um.edu.mt)

²Department of Artificial Intelligence, University of Malta, Msida, Malta (e-mail: dylan.seychell@um.edu.mt)

³Department of Physiology and Biochemistry, University of Malta, Msida, Malta (e-mail: claud.j.bajada@um.edu.mt)

Corresponding author: Neil Micallef (e-mail: neil.micallef.14@um.edu.mt).

The work discussed in this article is a summary of the findings from the corresponding author's dissertation project for his Master's in Artificial Intelligence with the University of Malta. The research work disclosed in this publication is partially funded by the Endeavour Scholarship Scheme (Malta). Scholarships are part-financed by the European Union - European Social Fund (ESF) - Operational Programme II - Cohesion Policy 2014-2020 "Investing in human capital to create more opportunities and promote the well-being of society".

ABSTRACT The accessibility and potential of deep learning techniques have increased considerably over the past years. Image segmentation is one of the many fields which have seen novel implementations being developed to solve problems in the domain. U-Net is an example of a popular deep learning model designed specifically for biomedical image segmentation, initially proposed for cell segmentation. We propose a variation of the U-Net++ model, which is itself an adaptation of U-Net, and evaluate its brain tumor segmentation capabilities. The proposed approach obtained Dice Coefficient scores of 0.7192, 0.8712, and 0.7817 for the Enhancing Tumor, Whole Tumor and Tumor Core classes of the BraTS 2019 challenge Validation Dataset. The proposed approach differs from the standard U-Net++ model in a number of ways, including the loss function, number of convolutional blocks, and method of employing deep supervision. Data augmentation and post-processing techniques were also implemented and observed to substantially improve the model predictions. Thus, this article presents a novel adaptation of the U-Net++ architecture, which is both lightweight, and performs comparably with peer-reviewed work evaluated on the same data.

INDEX TERMS Brain Tumor, BraTS, Deep Learning, Image segmentation, U-Net, U-Net++

I. INTRODUCTION

BRAIN tumors may be defined as abnormal growths of cells within the brain [1]. The 2020 Statistics for Adolescents and Young Adults [2] estimate 3700 cases of brain cancer, being the most common cause of death for men in this age group (10-39 years), and second largest cause of death overall after female breast cancer. The 2020 GLOBOCAN Cancer Statistics [3] estimate close to 19.3 million cancer cases worldwide, with close to 10 million deaths. Brain and nervous system cancers accounted for over 300,000 new cases, with 250,000 new deaths in 2020.

Magnetic Resonance Imaging (MRI) is a frequently used imaging method for diagnosing and monitoring brain tumors. The analysis of MR images may be categorized by the degree of user involvement. The work in [4] leverage this method of categorization and classifies techniques as being manual, semi-automatic, and fully-automatic brain tumor segmentation approaches.

A. MOTIVATION

According to [4], the clinical use of segmentation techniques generally depends on the simplicity of the approach and the level of interaction a system has with the user. Experts' level of trust in automated systems is another contributing factor. Thus, some medical institutions may favor manual segmentation over techniques which may appear complex and require extensive training. Manual brain tumor segmentation is a tedious process which requires analysts to manually trace the region of interest (ROI) on MR image slices, using software tools with sophisticated graphical user interfaces [4].

Manual segmentation is time consuming and also susceptible to human error such as inter and intra-operator variability, as shown in [5]. The latter work shows that maintaining a consistent manual segmentation strategy is difficult, even on the same MR image. Nonetheless, [6] claim that manual segmentation techniques are still carried out at a number of institutions. An automatic system for brain tumor segmen-

tation could minimize the drawbacks of human error and be invariant to external factors such as distractions and the mental state of the practitioner.

Current research has produced some capable automatic systems, as discussed in Section III-B. Thus, an individual developing an automatic segmentation system in present times should not focus solely on producing a model which learns the segmentation task and performs it automatically. Effort should also be investing in providing improvements such as adjusting the model's architecture to consume less resources, making it more accessible to practitioners and researchers alike.

Datasets for researching brain tumor segmentation have also become more widespread owing to competitions such as the Medical Image Computing and Computer Assisted Intervention (MICCAI) Multimodal Brain Tumor Segmentation Challenge (or BraTS) [7], [8], [9], [10], [11]. An example of a BraTS data sample and model prediction of the corresponding brain tumor are shown in Figure 1. Further explanation of BraTS and the BraTS datasets are provided in Section II-A.

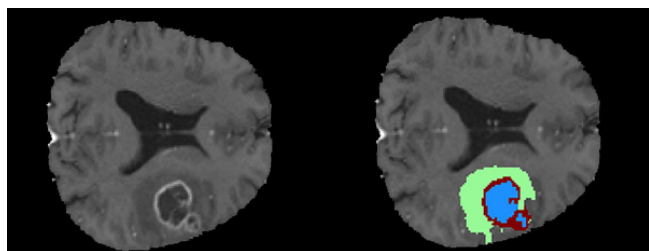


FIGURE 1. Left - BraTS sample, Right - Multiclass segmentation of the same sample, as predicted by the model presented in this paper.

B. AIM AND OBJECTIVES

The main aim of this paper is to create a model which takes multimodal 3D MR images as input to automatically generate a prediction of the corresponding brain tumor. The model output would also be compatible with standard MR viewers. This goal was achieved by following the below objectives:

- Surveying state-of-the-art methods at the time to produce a unique approach with results adequate for a clinical setting.
- Devising a model which works automatically, not requiring any user feedback or input for training and prediction.
- Adapting the U-Net++ [12] model architecture and identifying performance changes when modifying its features.

II. BACKGROUND

This section presents an outline of the data and the deep learning model architectures discussed in this article. The U-Net model [13] and its many adaptations are a considerable inspiration for the proposed model in this paper. One can also observe that the number of U-Net models submitted to the BraTS challenges increased significantly in recent years

of the challenge [14]. Thus, the background for each of the models presented in this work is also provided, followed by an introduction to the metrics used to evaluate each of the models.

A. MICCAI BRATS

The MICCAI BraTS challenge is a competition hosted by the Center for Biomedical Image Computing and Analytics (CBICA) at the University of Pennsylvania. The BraTS challenges identify and showcase state-of-the-art techniques for brain tumor segmentation. The datasets distributed by the competition organizers consist of real world data in the form of multi-institutional routine MRI scans, manually segmented by multiple board-certified neurologists [9].

The scans are split into high-grade gliomas (HGG) and low-grade gliomas (LGG) and provided in the T_1w , T_1ce , T_2w , and FLAIR modalities. The individual sequence types make the dataset more robust owing to the different strengths of each MR image modality. T_1 -weighted (or T_1w) sequences display fluid and water-based tissues as mid grey whilst fatty tissue has a high intensity [15]. Contrast agents applied to T_1w images produce T_1ce images, which enhance the intensity of highly vascular tumours [15].

T_2 -weighted (T_2w) images are visually opposite of T_1w scans, as fluids are now the brightest feature, and fat, water-based tissues are mid-grey [15]. Finally, FLAIR sequences are a variation of T_2w images, where the cerebrospinal fluid (CSF) within the brain and any tissues with a similar T_1 value are suppressed from the scan [15]. A sample of each sequence type taken from the training data used in this study is shown in Figure 2.

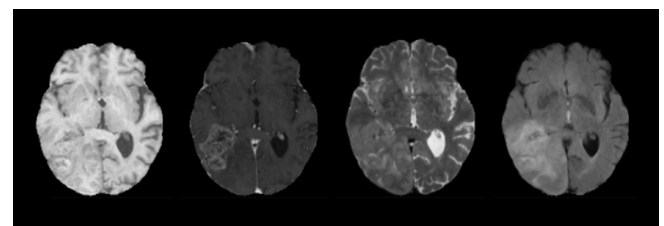


FIGURE 2. Left to Right: T_1 , T_1ce , T_2 , and FLAIR slice samples of the same patient from the BraTS 2019 training dataset

The BraTS data's multimodal nature allows competitors to devise segmentation approaches which are robust to the MRI sequence type. Since the data are also obtained from multiple institutions, this makes competition submissions also viable in real-world scenarios. In this paper, the 2019 challenge datasets were leveraged, as explained in Section IV-A.

B. U-NET AND RESIDUAL U-NET

Image segmentation problems present an additional layer of difficulty compared to more standard image/object recognition problems such as scene classification. In the latter problem, a model would learn to take images of scenery as input and produce one class label for the entire image. Predicting a class label for the entire scan would be sufficient only

e.g. when detecting whether an image contains a pathology, rather than identifying its location and extracting the tumor. In image segmentation, every pixel (or voxel for 3D images) will be assigned a class. This requires more complex feature extraction to be performed by a model. Moreover, due to the spatial resolution of these images, care must be taken not to encumber a network with too many parameters. This is the main inspiration behind the U-Net Convolutional Neural Network (CNN) [13].

The U-Net model is split into two halves, forming its synonymous ‘U-shape’. In the first half of the network (or ‘encoding’ path), an increasing amount of salient information from the input images is extracted at each level of the encoder. This is done by downsampling the input image and simultaneously doubling the size of the feature maps. The second half of the network (or ‘decoding’ path) performs the opposite function, restoring the size of the image whilst reducing the resolution of the feature maps.

Skip-connections connect both halves of the network via concatenation layers, which combine the information extracted from the encoding path with the data in the decoder. U-Net exhibited a model capable of performing biomedical image segmentation whilst maintaining a low number of parameters. The model performed well enough to achieve first place in the International Symposium on Biomedical Imaging (ISBI) challenge for segmentation of neuronal structures in electron microscopic stacks, by a considerable margin.

An important adaptation of the U-Net architecture is the residual U-Net. A notable variation was proposed by [16], who developed a U-Net model which used element-wise additions to combine the input and output of the convolutional blocks at each level of the first half of the network. The model also used small kernels and zero padding in its convolutions, and replaced max pooling with strided convolutions. Deep supervision [17], [18] was also employed in the decoder half of the network, where secondary segmentation maps were generated at each level of the decoder and combined using element-wise additions. Isensee *et al.* [19] would adapt [16] using a smaller batch size, double the filter map resolution, and a multi-class weighted Dice loss function as submissions for BraTS 2017 and BraTS 2018.

C. U-NET++

Another U-Net adaptation was proposed in [12], who proposed a model which made use of dense blocks within the U-Net architecture. The standard encoder-decoder structure of U-Net was maintained, however this was combined with additional upsampling layers along the skip-connections between the encoder and decoder halves of the network. This builds upon the convention of standard U-Net where a concatenation connects the encoder to the decoder at each level. The motivation behind this was to address the semantic gap between both halves of U-Net prior to concatenation [12]. The work by [20] combined U-Net++ and Half-Dense U-Net [21], which also shares properties of dense networks [22] and standard U-Net [13]. In [20], the combination of

both networks was done specifically to target difficulties in combining low-level and top-level features in convolutional neural networks.

The U-Net++ architecture allows for the concatenations to become increasingly refined at higher levels of the decoder part of the model. The standard U-Net architecture presented in [13] only upsamples layers from the decoder, following concatenation via skip-connection. U-Net++ maintains these layers and also includes further upsampling operations at every level of the first half of the network. This creates structures similar to smaller U-Nets within the model. The end result is the combination of U-Net’s architecture with more complex skip connections. In theory, this results in the combined benefit of lower parameters from the U-Net model with the rich feature space of dense networks. Moreover, [12] also made use of deep supervision along the first skip pathway, which produces full resolution segmentation maps.

Whilst the increased complexity of the model implies a correspondingly larger architecture, [12] claim that the number of parameters is quite similar to the original U-Net [13], and a wide variant of U-Net which uses larger feature channels. This comparison is also made on the grounds that the same number of convolutional kernels are used in both models. In [12], a comparison between U-Net++ and the standard, wide U-Net was computed using the Jaccard Index (also known as Intersection over Union or IoU). The resulting scores showed that U-Net++ outperformed standard and wide U-Net by an average of 2.8 to 3.3 points of IoU.

D. EVALUATION CRITERIA

The criteria used to assess the model’s performance closely follow the metrics used by the BraTS challenges. Namely, the predictions are evaluated on the basis of their Dice Coefficient, Sensitivity, Specificity, and Hausdorff Distance (95th percentile) across all three of the Whole Tumor (WT), Tumor Core (TC), and Enhancing Tumor (ET) target classes. The Dice Coefficient and Hausdorff Distance metrics calculate the model’s segmentation performance in terms of how closely the predicted tumor classes reflect the ground truth images. Regarding the use of the 95th percentile of the Hausdorff Distance, this was likely intended to avoid skewing the scores in case an extreme outlier exists in a model’s predictions. The sensitivity and specificity measurements calculate the capability of the model to minimize false negatives and false positives being predicted by the network.

There are a number of reasons behind the decision to keep these metrics as the final evaluation criteria for this paper. Firstly, the validation data are provided without ground truths, consisting only of the 125 multimodal patient volumes for BraTS 2019. Thus, evaluation is only possible on the CBICA BraTS web portal where a system impartially evaluates submissions against ground truths stored on the site. The portal then generates evaluation results make use of the aforementioned criteria. Secondly, this process provides a common framework for model evaluation which allows for

accurate comparison with other research, both for competition submissions and any alternative peer-reviewed works.

III. RELEVANT LITERATURE

This section will present a summarised timeline of brain tumor segmentation techniques, ranging from classical machine learning techniques such as clustering and support vector machines, to more modern approaches such as deep neural networks and U-Net adaptations. Particular emphasis was placed on techniques evaluated on different years of the BraTS challenges, especially for the deep learning approaches. The main reason for this is that it provides insight into how different methods performed on the BraTS data as a somewhat collective framework. Moreover, the data was becoming increasingly refined with every iteration of the challenge.

A. CLASSICAL MACHINE LEARNING TECHNIQUES

Initial research on brain tumor segmentation mainly consisted of several supervised and unsupervised machine learning (ML) approaches, as stated by [23]. Since datasets at the time were scarcer, data acquisition was more scattered, making it difficult to assess work since most studies would be using different datasets without a common evaluation technique. Nonetheless, unsupervised techniques proved useful in this time when unlabelled data was common as they did not rely on having high quality ground truth annotations accompanying an MR image dataset.

Clustering is one such technique which was frequently used for brain tumor detection and segmentation. The work by [24] explored the use of K-means clustering for tumor detection in MRI. The approach involved several stages, namely converting grayscale MR images to RGB, and then to CIELAB format, which makes use of chromaticity and luminosity coefficients. This approach was also used by [25] in their study using an ‘intuitionistic’ version of FCM. [26] later compared the performance of K-means, Fuzzy K-means, Gaussian Mixture Model (GMM), and Markov Random Field (MRF) on the GBM samples from the BraTS 2013 Test dataset. The best results for this study were obtained by the MRF approach, scoring 0.72, 0.62, and 0.59 Dice Coefficient scores for the WT, TC, and ET.

Supervised ML techniques such as Support Vector Machines (SVM) were also popularly used for brain tumor detection and segmentation. An example of SVM applied to this domain is the work by [27] who made use of a one-class SVM with an initial user seed point for the tumor used as input to the SVM, obtaining a percentage accuracy of 83.5% on 24 slices across 5 patients. A more recent approach by [28] used an SVM for feature extraction and classification in combination with FCM and PLTP. The work was evaluated on the BraTS 2013 and BraTS 2015 training datasets obtaining WT, TC, and ET Dice Scores of 0.76, 0.53, and 0.58 for 2013, and 0.81, 0.49, 0.47 on the 2015 dataset.

Random Forests are also a popular classifier for MR image segmentation. The implementation by [29] made use of a

GMM combined with a 2-stage Random Forest, obtaining Dice scores of 0.87 and 0.78 and 0.74 for the WT, TC, and ET for BraTS 2013. [30] later combined Random Forests with texture features for supervoxel classification, with positive results for BraTS 2013.

B. DEEP LEARNING TECHNIQUES

The popularity of classical machine learning methods and unsupervised approaches has waned in recent years, with the current trend shifting towards robust deep networks [23]. When examining submissions to the most recent iterations of the BraTS challenge, the main methods are mostly CNN variations, as showcased in [14]’s survey of BraTS competition submissions. In recent years, deep learning methods such as CNNs are preferred over the clustering and machine learning approaches seen in initial years of the competition.

A notable CNN implementation for brain tumor segmentation was proposed by [31], who made use of separate CNN pathways, one for HGG cases and the other for LGG, with different architectures and normalization configurations for each path. This research is also notable for its use of small convolutional kernels, inspired by [32]’s research on VGG-Nets. For image pre-processing, [33]’s bias field correction was implemented alongside an algorithm developed by [34] to standardise values across all sequences. [31] achieved first place in BraTS 2013 with WT, TC and ET Dice scores of 0.88, 0.83, 0.77, and second place in BraTS 2015 with Dice scores of 0.78, 0.65, and 0.75.

A 3D CNN for brain tumor segmentation named ‘DeepMedic’ was proposed by [35] for the 2015 and 2016 iterations of the BraTS challenge. The images were normalized by subtracting their mean and dividing by standard deviation. The CNN used was 11-layers deep, using two parallel-processing pathways at different resolutions. Small kernels were also used as in [31]. [35] also made use of residual connections in a new model extending DeepMedic, named ‘DMRes’. The performance of both models was evaluated on BraTS 2015 and 2016, and for 2015 DeepMedic obtained a Dice coefficient of 0.89, 0.75 and 0.72 for the WT, TC, and ET classes. DMRes performed better for the Dice and sensitivity metrics, but saw a slight decrease in precision. DMRes also achieved the top Dice scores for the TC and ET classes of images for the 2016 challenge, when combined with a Conditional Random Field approach.

[36] proposed an approach using deep neural networks for brain tumor segmentation. The architecture consisted of two pathways, making use of 7×7 and 13×13 feature map resolutions respectively. Bias-field correction and normalization were applied to the data for pre-processing. [36] also removed the top and bottom 1% of intensities from the input images. Training was also split into multiple phases to counter the healthy-to-diseased voxel imbalance, using a patch dataset with equiprobable labels. The project was evaluated on the BraTS 2013 test dataset, with competitive WT, TC, and ET Dice scores of 0.88, 0.79, and 0.73.

As discussed previously in Section II-B, two notable approaches using the residual U-Net architecture are the works by [16] and [19]. Both approaches were evaluated on separate BraTS datasets, obtaining very competitive results. Isensee *et al.* [19] also returned in 2018 with their ‘No New-Net’ [37] implementation. The latter work featured a very similar model to the 2017 submission, using a more refined pre-processing method and additional input data for training the model. No New-Net finished in second place for BraTS 2018.

Whilst the previous models made use of residual connections, [38] later made use of dense blocks [22] in a U-Net style network with encoding and decoding pathways. The work by [38] was an adaptation of the team’s previous semantic segmentation approach named ‘DeepSCAN’ [39]. The large parameter requirements of the dense DeepSCAN network was the motivation for [38] to integrate U-Net with the system, allowing for a lower spatial resolution within the dense portion of the network to keep the model size reasonable. This approach performed competitively in BraTS 2018, placing directly below No New-Net [37] in third place.

The model which secured first place in the BraTS 2018 challenge was proposed by [40]. The approach featured an encoder-decoder CNN with a variational auto-encoder (VAE) branch. This model works in a similar way to U-Net, with the main difference in this model being how the output of the encoder was split halfway into the mean and standard deviation, which were then used to generate samples from a Gaussian distribution to reconstruct the images prior to the beginning of the localisation process. The approach also used a very large patch size of $160 \times 192 \times 128$, which retained a large amount of the original images’ information. The Dice Coefficient scores obtained on the BraTS 2018 testing dataset for the WT, TC, and ET classes were 0.88, 0.82, and 0.77.

IV. METHODOLOGY

Prior to addressing each of the individual processes in the system pipeline, one can identify the entire workflow at a high level. The pipeline implemented and presented in this paper was adapted from a popular brain tumor segmentation online repository¹, aiming to replicate the implementation by [19]. An adequate understanding of the BraTS ground truth labels, target classes, and data distribution in terms of the HGG-to-LGG split is an essential complement to understanding these steps. Thus, a data definition section is provided prior to the breakdown of each step of the pipeline.

The first step in the pipeline involved pre-processing the input data using bias field correction, cropping, and normalization. With the data pre-processed and ready for training, the next step was to apply one-hot encoding to the ground truths. Once the model was trained on the input MR image volumes and ground truths, the model weights were preserved and used for generating predictions from the validation data, which had been passed through the pre-processing pipeline independently. The final step involved

resampling and interpolating the predictions to their initial dimensions, before uploading them to the BraTS web portal for the final evaluation. Each of these steps is explained further in the corresponding sections to follow.

A. DATA DEFINITION

The data for this paper were acquired from the 2019 MICCAI BraTS challenge. At the time of development, the 2019 data was the most robust version from all the challenge datasets, also including the largest amount of multi-institutional post-operative MRI scans. Moreover, an additional validation dataset was included with the training and testing datasets starting from BraTS 2017. It is of note that the BraTS 2019 testing dataset was unfortunately restricted to a 48-hour window during the live 2019 challenge, and not available for academic/research purposes. Nonetheless, since the validation data is an entirely separate dataset from that used for model training, it is valid for evaluation purposes.

The BraTS ground truth annotations are composed of three main categories, split into labels 1, 2, and 4 in the ground truths. The BraTS target classes ET, WT, and TC are composed of different combinations of these labels, as shown in Table I. A visual representation of the labels is also shown in Figure 3.

TABLE I. Labels and target classes for BraTS 2019

Class	Enhancing Tumor Core (Label 4)	Peritumoral Edema (Label 2)	Non-Enhancing & Necrotic Tumor Core (Label 1)
ET	✓		
WT	✓	✓	✓
TC	✓		✓

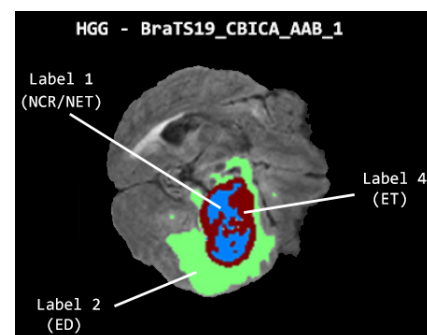


FIGURE 3. BraTS 2019 ground truth labels visualized. The blue mask is used for the non-enhancing and necrotic tumor core (label 1). Green is used for the peritumoral edema (label 2). Red is used for the enhancing tumor core (label 4).

The training data consist of 259 HGG and 76 LGG cases whilst the validation data consist of 125 cases which are not explicitly labelled as HGG or LGG. Examples of the HGG and LGG cases in the training data are shown in Figure 4. The

¹<https://github.com/ellisdg/3DUnetCNN>

image also shows the inter and intra-categorical differences for both LGG and HGG, exhibiting how even the same class of pathology can have varying shapes and textures. Both sets of the BraTS data are multimodal, consisting of the aforementioned T_1 , T_{1ce} , T_2 , and FLAIR sequence types. The data are available in a compressed Nifti (*.nii.gz) file format and categorized by case ID. Some samples were maintained from previous years, with all images manually segmented by multiple expert board-certified neuroradiologists [9].

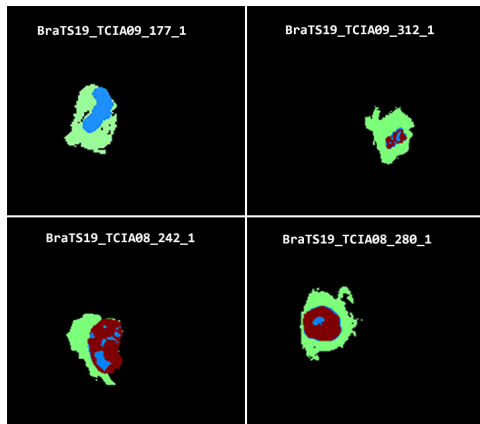


FIGURE 4. BraTS 2019 ground truth samples (LGG at the top, HGG at the bottom). One may observe how the pathologies vary visually even within the same class. Some LGG samples do not possess an ET segment.

B. PRE-PROCESSING

The input volumes were first passed through N4 bias field-correction [33], using the Advanced Normalization Tools (ANTs) library [41]. The FLAIR volumes were excluded from the bias-field correction process and included in the next pre-processing step with the corrected images. Background removal was then applied to each sample, removing all values between 0 and a relative tolerance parameter (in this case the default value of $1e^{-8}$). It should be noted that since each scan had a slightly different distribution of non-zero values, the cropping operation produced new images with different resolutions, which would not be viable for model training.

The images were thus resampled and interpolated to $128 \times 128 \times 128$, as in [19]. The image resizing steps were also applied to the corresponding ground truths, with the exception that nearest neighbour interpolation was used for the ground truths to avoid including values outside of the predefined BraTS labels. Finally, z-score normalization was used, transforming the input images to have zero mean and unit variance, using the formula in Equation (1), where x and x_{new} refer to the original and normalized samples, with μ and σ referring to the mean and standard deviation of the corresponding entire dataset.

$$x_{new} = \frac{x - \mu}{\sigma} \quad (1)$$

C. MODEL TRAINING

Following normalization, cropping, and resampling the images, the next step was training the model to automatically extract the multiclass tumor segments. Since the pipeline follows the process used in [19], the hyperparameters used during training were maintained. Samples were processed one-by-one rather than in batches due to the data's dimensionality. The ground truths were also passed through one-hot-encoding, transforming the original images with labels $\{1, 2, 4\}$ into multiple binary segmentation maps, i.e. one map with values $\{0, 1\}$ for each of the labels 1, 2, and 4.

The training dataset was split into an 80-20 train-test split, resulting in 268 total training steps. Each of the internal models (discussed in Section IV-D) were trained using these parameters, with the training period spanning 300 epochs and using a learning rate of $5e^{-4}$. The optimizer used for the model during training was the Adam gradient descent [42] optimizer. To handle the class imbalances present in the data, the multi-class adaptation of the Dice loss devised by [19] was used, as presented in Equation (2).

$$L = -\frac{1}{K} \sum_{k \in K} \frac{2(Y_k \cap \hat{Y}_k) + \alpha}{(Y_k + \hat{Y}_k) + \alpha} \quad (2)$$

Here, K refers to the 3 ground truth labels and Y , \hat{Y} refer to the images of the ground truth and model prediction respectively. The divisor coefficient and summation outside of the main function modifies the standard Dice loss to handle multiclass evaluation, and α refers to a smoothing constant with a value of $1e^{-5}$. One should note that training was largely carried out on Google Cloud and split between two server instances. The initial machine made use of a Tesla K80 GPU with 12 GB of virtual memory. A switch was made shortly after to an instance with a Tesla P100 GPU with 16GB of virtual memory. Apart from the increased GPU memory, the compute capability of the Tesla P100 was much higher, allowing for training to complete much faster. Training the final model for 300 epochs took anywhere between 2 to 3 days when using the Tesla P100 GPU, compared to the 6 to 7 day duration when using the Tesla K80. More detailed parameters related to training are shown in Table II. The tabulated data includes information such as the amount and type of GPU memory, CUDA cores, and exact training times for all of the hardware used for model training at different stages of this implementation.

TABLE II. GPU and training time comparisons for the MELECON U-Net++ model, discussed further in Section V-A2. This model was runnable on a GTX970 GPU (local hardware). Training time for the other models is slightly higher, but only by a small margin.

GPU	CUDA	CUDA cores	GPU memory	T/step (s)	T/epoch (s)	T/300 epochs (hrs)
GTX 970M	5.2	1,280	6 GB GDDR5	2	589	49
Tesla K80	3.7	4,992	12 GB GDDR5	8	2,185	182
Tesla P100	6.0	3,584	16 GB HBM2	2	520	43

Data augmentation was also applied during training to produce synthetic samples of the BraTS training images. As

stated by [13], the objective of using data augmentation for datasets with limited data is to produce a more robust dataset for the model during training. For this experiment, random permutations of rotations, axes flips, and transpositions were applied to the training batches. Rotations were applied to the images in multiples of 90 degrees, and axes flips were performed on all three of the x, y , and z axes. Transposition in this case refers to the image data matrix being transposed, changing the order of the dimensions.

D. MODELS

Three U-Net based models were built internally following the training process described in Section IV-C. The first of these models takes inspiration from the original U-Net model [13], shown in Figure 5. The encoder part of the standard U-Net model features convolutional blocks composed of two $3 \times 3 \times 3$ convolutions with a standard ReLU nonlinearity function followed by a $2 \times 2 \times 2$ max pooling operation. The small convolutional kernels allow the model to maintain a relatively small number of parameters [32]. These models were built to compare U-Net, Residual U-Net, and the proposed model within the same data processing pipeline and training conditions. This comparison between the models is described further in Section V-C.

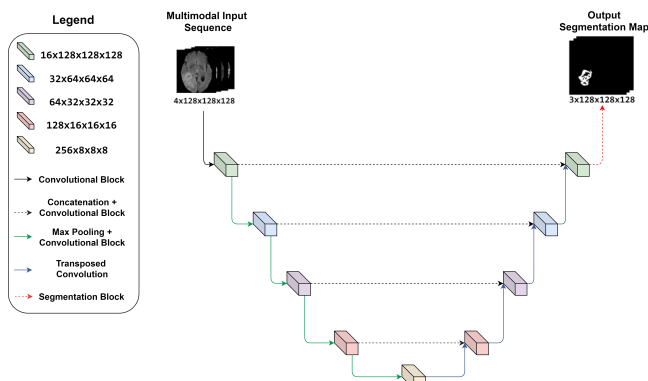


FIGURE 5. U-Net model built internally for the implementation presented in this paper.

There are five levels of depth in the network, with the final level being a bridge to the decoder part of the network. Concatenation layers connect both halves of the model at each level apart from the deepest block. Following initial experiments showing that dropout layers with the tested value were not beneficial to the approach, they were omitted from the standard U-Net model.

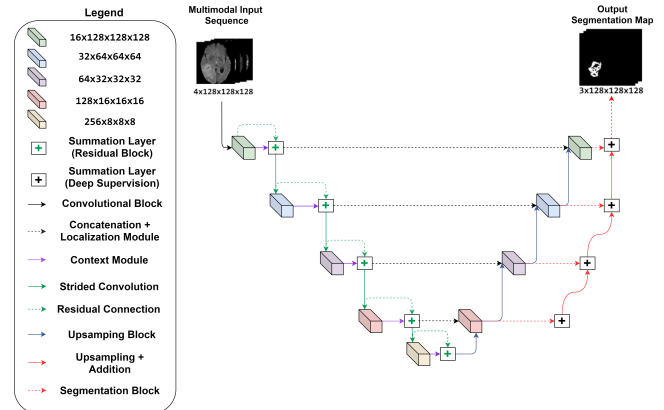


FIGURE 6. Residual U-Net model built internally for the implementation presented in this paper.

The second internal model follows the residual U-Net architecture devised by [19], shown in Figure 6. This model was built as per the aforementioned Github repository², to be consistent with the work in [19]. Some differences to the standard U-Net are the addition of residual blocks and the use of strided convolutions in place of max pooling along the encoder. Although there is no empirical evidence proving that strided convolutions are always superior to max pooling, it introduces the possibility for the model to ‘learn’ how to downsample the images better.

The residual U-Net model also uses upsampling layers in place of transposed convolutions in the decoder as [19] claim that the latter may produce checkerboard artifacts in the output. The model also makes use of deep supervision, with secondary segmentation maps being generated along the decoder half of the network, using element-wise additions. The objective of this approach is to refine the final segmentation predictions generated by the model. The final, and proposed model is an adaptation of U-Net++ [12], shown in Figure 7.

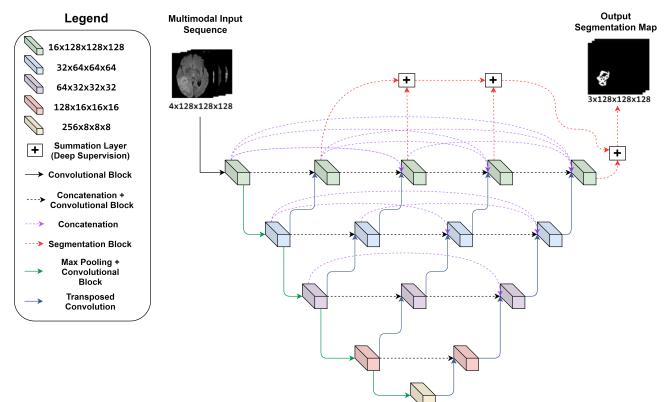


FIGURE 7. U-Net++ model built internally for the implementation presented in this paper.

One can observe how the main difference between this model and the standard U-Net architecture is the more complex system of skip connections. Upsampling layers are now

²<https://github.com/ellisdg/3DUnetCNN>

also present in the encoder part of the network in U-Net++, propagating information from deeper parts of the encoder up to the topmost layers. Moreover, deep supervision is also present here, however, this time it is placed along the first skip connection. The benefit of this approach with U-Net++ is that the blocks along the first concatenation produce full-resolution segmentation maps, consisting of upsampled feature data from the deeper layers of the encoder.

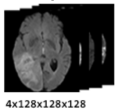
Since the U-Net++ model is a convolutional neural network, the model parameters learning during training are generated by the 3D convolutional layers, instance normalization, and transposed convolutions. The number of parameters per convolutional layer (standard and transposed) is calculated using the formula in Equation 3:

$$p = ((x * y * z * d) + 1) * k \quad (3)$$

Where x , y , and z refer to the convolutional kernel parameters ($3 \times 3 \times 3$). d refers to the number of filters in the previous layer, and k refers to the number of filters in the current layer. Figure 8 shows how Equation 3 applies to the proposed model.

Model Parameters Calculation			
Section	Layers	Output Shape	Parameters
1. Conv Block	a. Conv3D	a. $16 \times 128 \times 128 \times 128$	a. $((3 \times 3 \times 3 \times 4) + 1) \times 16 = 1744$
	b. Instance Normalization	b. $16 \times 128 \times 128 \times 128$	b. $16 \times 2 = 32$
	c. Activation	c. $16 \times 128 \times 128 \times 128$	c. 0
2. Max Pooling + Conv Block	a. MaxPooling3D	a. $16 \times 64 \times 64 \times 64$	a. 0
	b. Conv3D	b. $32 \times 64 \times 64 \times 64$	b. $((3 \times 3 \times 3 \times 16) + 1) \times 32 = 13856$
	c. Instance Normalization	c. $32 \times 64 \times 64 \times 64$	c. $32 \times 2 = 64$
	d. Activation	d. $32 \times 64 \times 64 \times 64$	d. 0
3. Transposed Convolution	a. Transposed Convolution3D	a. $16 \times 128 \times 128 \times 128$	a. $((3 \times 3 \times 3 \times 32) + 1) \times 16 = 13840$
4. Concatenation + Conv Block	a. Concatenation	a. $32 \times 128 \times 128 \times 128$	a. 0
	b. Conv3D	b. $16 \times 128 \times 128 \times 128$	b. $((3 \times 3 \times 3 \times 32) + 1) \times 16 = 13840$
	c. Instance Normalization	c. $16 \times 128 \times 128 \times 128$	c. $16 \times 2 = 32$
	d. Activation	d. $16 \times 128 \times 128 \times 128$	d. 0

Multimodal Input Sequence



$4 \times 128 \times 128 \times 128$

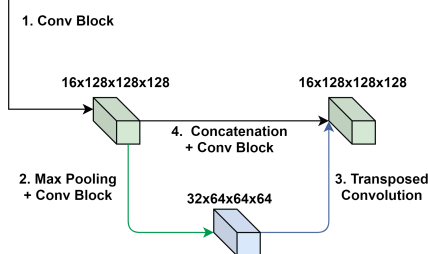


FIGURE 8. An example of model parameter calculation, shown on a segment of the proposed model.

Combining this information with the model structure shown in Figure 7, the total amount of model parameters increases proportionally with the number of filters. This is primarily reflected in deeper layers in the model, and concatenation layers, both of which have outputs with larger filter sizes. Taking all of the above into consideration, the proposed model's total number of parameters 4,516,7000.

Whilst this proposed model is heavily inspired by the U-Net++ in [12], there are a number of key differences in the approach presented in this article. One of the principal differences is the convolutional block schema used by the proposed model. The original U-Net++ by [12] uses a horizontal block scheme which resembles the standard U-Net model, with two sets of convolution, batch normalization, and ReLU activations. Following the experiment described in Section V-A4, it was discovered that halving the number of convolutional blocks resulted in comparable results. The main benefit from this experiment was that the number of U-Net++ parameters using our setup dropped from 7.7M to 4.5M. Furthermore, the entire original U-Net++ model architecture presented in [12] totalled 9.04M model parameters. The drop in parameters is substantial, as smaller models with a lesser total of model parameters are less likely to overfitting to the input data during training.

Other differences include the loss function, explored in Section V-A3. Our model uses the weighted multi-class Dice Coefficient loss implemented by [19] in their Residual U-Net implementation, rather than the composite binary-crossentropy Dice function used by [12]. The means of implementing 'deep supervision' to refine the secondary segmentation maps also differs from [12]'s averaging or fast-selection approaches. In the U-Net++ model proposed in this article, the secondary segmentation maps are actually combined using element-wise additions, as shown in Figure 7. Initial training runs showed that the model's convergence improved greatly when comparing the model with and without the element-wise additions for the segmentation maps.

Other differences in our approach include the use of instance normalization, as our model only processes 'batches' of individual patients, hence batch normalization would destabilize training. The model also does not make use of dropout layers, and use a starting filter map resolution of 16 rather than 32 as in [12]. Moreover, the convolutional kernels used for segmentation have a resolution of $3 \times 3 \times 3$ rather than $1 \times 1 \times 1$. Whilst this provided only minor improvements in initial training runs, this was maintained for subsequent training of the model.

V. EVALUATION

This section will serve to exhibit the proposed model's performance. A number of experiments were conducted to extend the model and training parameters, and identify any possible improvements to the final results. A detailed description of each experiment is provided in the sections to follow, including a summary of all experiments conducted in this research effort. Following the best model configuration being

selected, the final results on the BraTS 2019 validation data were obtained. The results were compared internally with a standard U-Net inspired by the work in [13] and a residual U-Net model [19] architecture. An external evaluation was also conducted against a number of peer-reviewed approaches on the BraTS 2019 validation data.

A. EXPERIMENTS

1) Ablation Study - Data Augmentation

Data augmentation techniques are used to generate synthetic samples of real-world data to create more input samples for model training. This is generally helpful for training models tasked with solving problems with scarce data, such as biomedical image segmentation. The original U-Net [13] proposal also made use of data augmentation techniques in this regard. To assess whether or not data augmentation was being beneficial to the final model predictions, an ablation study was conducted, comparing two separate training runs. The results are presented in Tables III and IV.

TABLE III. Dice Coefficient and Hausdorff Distance comparison between the proposed U-Net++ without and with data augmentation on the BraTS 2019 Validation Dataset. Best scores in bold.

Configuration	Dice Coefficient			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC
U-Net++ (no Data Augmentation)	0.6505	0.8474	0.7285	8.2741	11.9739	9.6433
U-Net++ (with Data Augmentation)	0.6920	0.8709	0.7824	6.8001	8.3279	9.4997
Average Improvement	5.52%			16.58%		

TABLE IV. Sensitivity and Specificity comparison between the proposed U-Net++ without and with data augmentation on the BraTS 2019 Validation Dataset. Best scores in bold.

Configuration	Sensitivity			Specificity		
	ET	WT	TC	ET	WT	TC
U-Net++ (no Data Augmentation)	0.7080	0.8606	0.7078	0.9979	0.9922	0.9971
U-Net++ (with Data Augmentation)	0.7208	0.8654	0.7655	0.9979	0.9945	0.9969
Average Improvement	3.51%			0.07%		

Comparing the tabulated scores, one can observe how data augmentation led to a substantial improvement across all categories of the evaluation criteria. The improved Dice Coefficient and Hausdorff Distance scores show that the segmentation performance of the model improved greatly when implementing data augmentation. This may be attributed to the fact that the new synthetic samples generated during training allowed the model to generalise better, improving tumour segmentation on the unseen validation data. The increase in sensitivity shows that the model also performed better in terms of avoiding false negatives. The marginal increase in average sensitivity score may be attributed to the fact that the initial score obtained by the model was already very high.

2) Using Upsampled Features Directly in Skip-Connections

The next set of evaluated models were more compact versions of the proposed U-Net++ model shown in Section IV-D. The networks were work-in-progress models being tested on local hardware. Thus, some minor modifications to the architecture were made to fit the networks on 6GB of GPU memory. These models follow the proposed U-Net++ architecture closely with two minor differences: a) features upsampled from the encoder were concatenated directly along the skip-connection rather than being passed through a convolutional block and additional concatenation; b) only the penultimate secondary segmentation map was used in the element-wise additions to refine the final segmentation result via deep supervision.

Two variations of this model were created. One was trained for 100 epochs as a part of research submitted to the Organization for Human Brain Mapping (OHBM) 2020 Annual Meeting [43]. The other model was trained for 300 epochs and submitted to the IEEE Mediterranean Electrotechnical Conference (MELECON 2020) Conference [44]. In spite of being simpler variations of the proposed model and being evaluated on a holdout set of the BraTS 2019 training data, both models were accepted by the respective bodies. In this experiment, we compared the results of these models against the final, proposed U-Net++ on the BraTS 2019 validation data, shown in Tables V and VI.

TABLE V. Dice Coefficient and Hausdorff Distance comparison between the proposed U-Net++ and conference models on the BraTS 2019 Validation Dataset. Best scores in bold.

Configuration	Epochs	Parameters	Dice Coefficient			Hausdorff Distance		
			ET	WT	TC	ET	WT	TC
U-Net++ (OHBM)	100	4.4M	0.6510	0.8642	0.7202	8.4248	8.4060	10.6127
U-Net++ (MELECON)	300	4.4M	0.6711	0.8631	0.7592	7.3657	9.0543	10.0087
U-Net++ (Proposed)	300	4.5M	0.6920	0.8709	0.7824	6.8001	8.3279	9.4997

TABLE VI. Sensitivity and Specificity comparison between the proposed U-Net++ and the conference models on the BraTS 2019 Validation Dataset. Best scores in bold.

Configuration	Epochs	Parameters	Sensitivity			Specificity		
			ET	WT	TC	ET	WT	TC
U-Net++ (OHBM)	100	4.4M	0.7277	0.8452	0.6772	0.9970	0.9954	0.9977
U-Net++ (MELECON)	300	4.4M	0.6804	0.8606	0.7445	0.9980	0.9940	0.9967
U-Net++ (Proposed)	300	4.5M	0.7208	0.8654	0.7655	0.9979	0.9945	0.9969

The proposed model outperformed both of the other approaches in the majority of the criteria, particularly for the Dice Coefficient. This is the expected outcome seeing as the proposed model is the 'full' version of U-Net++, leveraging the entire arsenal of dense connections and all secondary segmentation maps for deep supervision. This is also the reason why both of the conference models in this comparison have a very slightly lesser amount of parameters. In addition, whilst the OHBM model obtained a slightly higher sensitivity score for the enhancing tumour class, the scores for the whole tumour and tumour core were much less than those of the proposed model.

3) Optimization Function

The next experiment evaluates the function used to optimize the model's training. In this paper, the employed loss function follows the multiclass Dice Coefficient loss proposed by [19] and shown in Equation (2). Nonetheless, since the proposed model is not a residual U-Net as in [19], we decided to also attempt training the model using a function which follows the binary cross-entropy loss used by [12] in the original U-Net++ paper, shown in Equation (4),

$$L = 0.5 * BCE - DSC$$

$$L = 0.5 * \frac{1}{K} \sum_{k \in K} (Y_k * \log(\hat{Y}_k) + (1 - Y_k) * \log(1 - \hat{Y}_k)) - DSC$$

$$L = 0.5 * \frac{1}{K} \sum_{k \in K} (Y_k * \log(\hat{Y}_k) + (1 - Y_k) * \log(1 - \hat{Y}_k)) - \frac{1}{K} \sum_{k \in K} \frac{2(Y_k \cap \hat{Y}_k) + \alpha}{(Y_k + \hat{Y}_k) + \alpha} \quad (4)$$

where BCE and DSC refer to the binary cross-entropy and standard Dice Coefficient function. Y , \hat{Y} refer to the BraTS ground truth and model prediction. K refers to the set of target classes and α is a smoothing constant with a value of $1e^{-5}$. The comparison between both optimization functions is shown in Tables VII and VIII.

TABLE VII. Dice Coefficient and Hausdorff Distance comparison between the proposed Dice Loss and the original binary crossentropy composite(BCE) loss functions on the BraTS 2019 Validation Dataset. Best scores in bold.

Model	Dice Coefficient			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC
U-Net++ (Proposed Dice Loss)	0.6920	0.8709	0.7824	6.8001	8.3279	9.4997
U-Net++ (BCE Dice Loss)	0.6876	0.8616	0.7656	7.5247	9.5144	11.5140
Average Improvement	-1.28%			-15.46%		

TABLE VIII. Sensitivity and Specificity comparison between the proposed Dice Loss and the original binary crossentropy composite(BCE) loss functions on the BraTS 2019 Validation Dataset. Best scores in bold.

Model	Sensitivity			Specificity		
	ET	WT	TC	ET	WT	TC
U-Net++ (Proposed Dice Loss)	0.7208	0.8654	0.7655	0.9979	0.9945	0.9969
U-Net++ (BCE Dice Loss)	0.7278	0.8925	0.7910	0.9979	0.9912	0.9940
Average Improvement	2.48%			-0.21%		

From the results, one may notice that the Dice optimization function was superior in terms of raw segmentation, i.e. the Dice Coefficient and Hausdorff Distance. The intuition behind this result is that the use of the multiclass Dice Coefficient function in the proposed model allowed for a better overall classification of the tumour segments. Conversely, the binary cross-entropy loss performed better in terms of sensitivity and specificity. We followed the same route as many other works (such as [45], [46]), who prioritise the Dice Coefficient when evaluating models using BraTS data. As a result of this, the weighted multi-class Dice Coefficient function was kept for the proposed model.

4) Using Original U-Net++ Convolutional Blocks

Research such as [36] claims that in some instances, adding additional convolutional blocks or increasing the filter map resolution did not result in any substantial performance increase in their CNN models. When testing different iterations of the model, one of the main considerations taken into account was the size of the model, in this case the number of model parameters. This was also highlighted by the very long training times for each of the internal models, as shown in Table II. Taking all of the above factors into consideration, it was decided to test the model using only half of the convolution-normalization-activation blocks as in the original work by [12]. In essence, the goal was to check whether the tradeoff between model parameters and performance would be worth pursuing. The results are shown in Tables IX and X.

TABLE IX. Dice Coefficient and Hausdorff Distance comparison between the proposed and original U-Net++ convolutional block schema on the BraTS 2019 Validation Dataset. Best scores in bold.

Configuration	Parameters	Dice Coefficient			Hausdorff Distance		
		ET	WT	TC	ET	WT	TC
U-Net++ (Proposed Conv Blocks)	4.5M	0.6920	0.8709	0.7824	6.8001	8.3279	9.4997
U-Net++ (Original Conv Blocks)	7.7M	0.6931	0.8690	0.7778	5.2130	7.6872	9.6055
Average Improvement	-69.38%	-0.22%			9.97%		

TABLE X. Sensitivity and Specificity for U-Net++ comparison between the proposed and original U-Net++ convolutional block schema, on the BraTS 2019 Validation Dataset. Best scores in bold.

Configuration	Parameters	Sensitivity			Specificity		
		ET	WT	TC	ET	WT	TC
U-Net++ (Proposed Conv Blocks)	4.5M	0.7208	0.8654	0.7655	0.9979	0.9945	0.9969
U-Net++ (Original Conv Blocks)	7.7M	0.6893	0.8912	0.7957	0.9984	0.9922	0.9956
Average Improvement	-69.38%	0.85%			-0.10%		

From the results obtained, we can see that the two models obtain near equivalent results, barring the Hausdorff distance measurement. Conversely, the proposed model with the lesser number of parameters obtained a slightly improved average Dice Coefficient. These two results combined infer that the proposed model had a larger segmentation error for the 'worst' occurrence, yet still performed slightly better than the larger model on average, as shown by the Dice Coefficient. In our opinion, the 69% reduction in model parameters of the proposed model is more significant than the minor decrease in Hausdorff Distance and average sensitivity score. Thus, the new block schema with the lesser amount of parameters was maintained.

5) Ablation Study - Dropout Regularisation

Dropout regularisation is commonly used in CNNs, in an attempt to reduce the possibility of the model overfitting to the training data. The latter process causes the model to only learn the salient features from the training data, rather than being able to generalise for new, unseen samples. In this

experiment, we used the original online repository's dropout value of 0.3, with the results shown in Tables XI and XII.

TABLE XI. Dice Coefficient and Hausdorff Distance comparison between the proposed U-Net++ with and without dropout regularization on the BraTS 2019 Validation Dataset. Best scores in bold.

Model	Dice Coefficient			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC
U-Net++ (no Dropout)	0.6920	0.8709	0.7824	6.8001	8.3279	9.4997
U-Net++ (with Dropout)	0.6940	0.8589	0.7684	7.7722	8.3574	9.7855
Average Improvement	-0.95%			-5.89%		

TABLE XII. Sensitivity and Specificity comparison between the proposed U-Net++ with and without dropout regularization on the BraTS 2019 Validation Dataset. Best scores in bold.

Model	Sensitivity			Specificity		
	ET	WT	TC	ET	WT	TC
U-Net++ (no Dropout)	0.7208	0.8654	0.7655	0.9979	0.9945	0.9969
U-Net++ (with Dropout)	0.7378	0.8653	0.7603	0.9978	0.9933	0.9968
Average Improvement	0.56%			-0.05%		

The results for this particular dropout value show that there was no substantial improvement in terms of model prediction. For this reason, we decided to not use dropout regularization going forward. In our case, experiment prioritisation is the main reason for only having a singular dropout test using a value of 0.3. Thus, additional testing with other dropout values is encouraged, as it may lead others to obtain more positive results. This is also mentioned in Section VI-B.

6) Post-Processing Analysis

The final set of experiments relate to possibilities of improving the model's predictions after training. For every set of predictions uploaded to the CBICA BraTS web portal, a spreadsheet containing the evaluation scores for each patient is provided to the uploader. Some of the result files extracted for previous experiments showed patients with an ET Dice Coefficient score of 0, as shown in Table XIII.

TABLE XIII. BraTS 2019 Validation Dataset cases with ET Dice score of 0.

Label	Dice Coefficient			NET	Voxels	
	ET	WT	TC		ET	Edema
BraTS19_TCIA09_248_1	0	0.9210	0.5672	16403	83	41162
BraTS19_TCIA10_127_1	0	0.8941	0.8713	10741	19	4975
BraTS19_TCIA10_195_1	0	0.9476	0.7854	50453	1514	82176
BraTS19_TCIA10_232_1	0	0.8965	0.6610	68831	173	71083
BraTS19_TCIA10_609_1	0	0.9514	0.9200	47509	7	25112
BraTS19_TCIA10_614_1	0	0.9301	0.3670	4187	73	21716
BraTS19_TCIA11_612_1	0	0.8155	0.8531	9314	0	14084
BraTS19_TCIA13_619_1	0	0.9033	0.7342	17653	0	62305
BraTS19_TCIA13_648_1	0	0.7823	0.6530	38882	0	26505
BraTS19_TCIA13_652_1	0	0.9258	0.1222	12362	0	19084

A thorough analysis was conducted on the patients with ET Dice Scores of this nature, elaborated further in Section VII-A below. Following the correct criteria for post-processing being identified, the final step was to confirm that the positive scores obtained via post-processing would

not serve to diminish any of the other scores. This test was conducted by comparing the quality of the predictions with and without zero thresholding, shown in Tables XIV and XV.

TABLE XIV. Dice Coefficient and Hausdorff Distance comparison between the proposed U-Net++ with and without post-processing on the BraTS 2019 Validation Dataset. Best scores in bold.

Variation	Dice Coefficient			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC
U-Net++ (no Post-processing)	0.6920	0.8709	0.7824	6.8001	8.3279	9.4997
Post-processing (Ratio)	0.7113	0.8709	0.7824	5.2052	8.3330	9.4990
Post-processing (Constant)	0.7192	0.8712	0.7817	4.6861	8.2157	9.4748
Average Improvement (Ratio)	0.93%			7.80%		
Average Improvement (Constant)	1.29%			10.90%		

TABLE XV. Sensitivity and Specificity comparison between the proposed U-Net++ with and without post-processing on the BraTS 2019 Validation Dataset. Best scores in bold.

Variation	Sensitivity			Specificity		
	ET	WT	TC	ET	WT	TC
U-Net++ (no Post-processing)	0.7208	0.8654	0.7655	0.9979	0.9945	0.9969
Post-processing (Ratio)	0.7248	0.8653	0.7655	0.9979	0.9945	0.9969
Post-processing (Constant)	0.7232	0.8671	0.7630	0.9980	0.9944	0.9969
Average Improvement (Ratio)	0.18%			0%		
Average Improvement (Constant)	0.07%			0%		

As expected, the main improvement from this experiment was for the enhancing tumour category, since the post-processing pipeline was built to handle patient cases with an ET score of 0. The recorded improvements are particularly substantial for the Dice Coefficient and Hausdorff Distance, with the best results overall being obtained by the constant threshold post-processing approach, which was thus maintained for the final model.

7) Summary of Experiments

This section presents all of the results obtained from the experiments performed in this paper. The harmonised results for all of the experiments discussed in this section are shown in Tables XVI and XVII.

TABLE XVI. Dice Score and Hausdorff Distance for all experiments performed for this paper. All models after the first make use of data augmentation. 'Baseline' refers to the proposed U-Net++ without post-processing. Proposed model configuration and best scores in bold.

Configuration	Dice Coefficient			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC
No Data Augmentation	0.6505	0.8474	0.7285	8.2741	11.9739	9.6433
OHBM	0.6510	0.8642	0.7202	8.4248	8.4060	10.6127
MELECON	0.6711	0.8631	0.7592	7.3657	9.0543	10.0087
BCE Dice Loss	0.6876	0.8616	0.7656	7.5247	9.5144	11.5140
Double Conv. Blocks	0.6931	0.8690	0.7778	5.2130	7.6872	9.6055
Dropout	0.6940	0.8589	0.7684	7.7722	8.3574	9.7855
Baseline	0.6920	0.8709	0.7824	6.8001	8.3279	9.4997
Baseline - Ratio Thresh. (0.04)	0.7113	0.8709	0.7824	5.2052	8.3330	9.499
Baseline - Constant Thresh. (200)	0.7192	0.8712	0.7817	4.6861	8.2157	9.4748

TABLE XVII. Sensitivity and Specificity for all experiments performed for this paper. All models after the first make use of data augmentation. 'Baseline' refers to the proposed U-Net++ without post-processing. Proposed model configuration and best scores in bold.

Configuration	Sensitivity			Specificity		
	ET	WT	TC	ET	WT	TC
No Data Augmentation	0.7080	0.8606	0.7078	0.9979	0.9922	0.9971
OHBM	0.7277	0.8452	0.6772	0.9970	0.9954	0.9977
MELECON	0.6804	0.8606	0.7445	0.9980	0.9940	0.9967
BCE Dice Loss	0.7278	0.8925	0.7910	0.9979	0.9912	0.9940
Double Conv. Blocks	0.6893	0.8912	0.7957	0.9984	0.9922	0.9956
Dropout	0.7378	0.8653	0.7603	0.9978	0.9933	0.9968
Baseline	0.7208	0.8654	0.7655	0.9979	0.9945	0.9969
Baseline - Ratio Thresh. (0.04)	0.7248	0.8653	0.7655	0.9979	0.9945	0.9969
Baseline - Constant Thresh. (200)	0.7232	0.8671	0.7630	0.9980	0.9944	0.9969

Going through each of the experiments sequentially, the data augmentation was undoubtedly one of the larger improvements applied to the proposed model. The conference models (OHBM and MELECON) exhibited slightly lower scores, mostly owing to the fact they were lesser versions of the proposed U-Net++. The binary cross-entropy Dice optimization function implemented in the original U-Net++ by [12] exhibited higher sensitivity scores, yet showed lesser Dice Coefficient and Hausdorff Distance scores when compared to the proposed approach. The convolutional block schema implemented by [12] was also not favoured over the proposed structure, as this provided only marginal improvements at the cost of 69.38% increased model parameters.

The dropout experiment also showed no notable improvements to the overall segmentation performance of the model. Having said this, it could be beneficial to perform further testing with different dropout values. Finally, the post-processing experiment was successful, as the constant voxel thresholding served to improve the model's enhancing tumour segmentation without diminishing performance in other metrics. Following the observations noted in this section, as well as the prioritization of the Dice Coefficient as the main criteria for evaluation, the proposed U-Net++ maintains the data augmentation and post-processing pipelines, the multiclass Dice Coefficient optimization, and lesser amount of convolutional blocks.

B. RESULTS

The BraTS 2019 validation dataset was used to assess the model's performance. The final scores averaged over all 125 patient samples are shown in Table XVIII.

TABLE XVIII. Mean results for the final proposed model on the BraTS 2019 Validation Set, obtained from the CBICA IPP.

Metric	ET	WT	TC
Dice Coefficient	0.7192 ± 0.2811	0.8712 ± 0.0934	0.7817 ± 0.1914
Sensitivity	0.7232 ± 0.2941	0.8671 ± 0.0965	0.7631 ± 0.1997
Specificity	0.9980 ± 0.0045	0.9944 ± 0.0063	0.9969 ± 0.0059
Hausdorff Distance	4.6863 ± 6.5129	8.2157 ± 9.8122	9.4752 ± 12.3579

As previously mentioned, [36] discovered that from the 2% of pathological pixels in the scan, over half of the distribution were edema pixels. From the results obtained in Table XVIII, the scores obtained in the WT category also reinforce this.

The whole tumour obtained the highest Dice Coefficient and Sensitivity scores by a wide margin, and it also the only target class containing the edema tumour section. This is a pattern which is observable throughout other research evaluated on the BraTS datasets. Conversely, the presence of LGG patient cases without a tumour segment and low representation of the ET tumour section in the data may be contributors to the lower scores obtained for this class. LGG's may also be difficult to classify for a model since they have less than 25% representation in the dataset compared to HGG subjects. Box and whisker plots for the Dice Coefficient and Hausdorff Distance are shown in Figure 9, giving a deeper look into the scores obtained on the evaluation data.

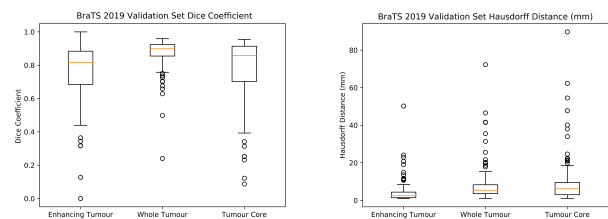


FIGURE 9. Box and whisker plots for the 2019 BraTS validation set Dice Coefficient and Hausdorff Distance.

Observing the Dice Coefficient results shown in Figure 9, the whole tumour is once again shown to be well represented, and with minor variance compared to the enhancing tumour and tumour core. The outliers are spread out for all three classes, with the ET segments having the most significant outliers, owing to the known cases with an ET Dice Coefficient of 0, previously discussed in Section VII-A. The median Dice scores for each class are above 0.8. Since the Dice Coefficient represents the segmentation accuracy between the model predictions and ground truths, these values exhibit that the median segmentation performance was a fairly high number.

One observation when comparing the box plots for the Dice Coefficient and Hausdorff Distance is that the distributions for ET and WT change considerably. Nonetheless, these changes and the larger amount of outliers could partially be attributed to the nature of the measure which takes into account the 95th percentile of the largest segmentation error. This is also substantiated by the WT and TC having long whiskers, which suggests that the range of Hausdorff values varies greatly in both cases. The interquartile range for both the Dice Coefficient and the Hausdorff Distance are fairly well contained, which implies that the results are reliable.

Since the ground truths for the validation data are kept on CBICA IPP and not distributed to competitors, it is not possible to visualize outliers directly on MR images. Nonetheless, analysis for correlations may still be carried out from the output files produced by the IPP. Outlier observations for the Dice Coefficient and Hausdorff Distance are shown in Figure 10. The whole tumour is used as an example, as it was found to have the most non-zero outliers.

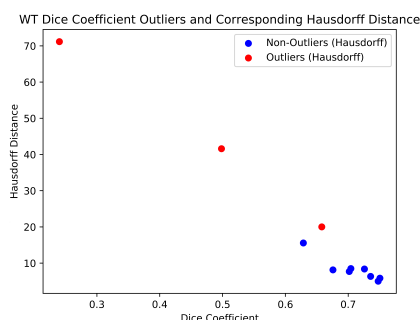


FIGURE 10. Scatter plot showing validation samples with an outlying WT Dice Coefficient and corresponding Hausdorff Distance.

Figure 10 shows that the values of Dice Coefficient outliers vary fairly proportionally with the corresponding Hausdorff Distance. The other observation from the plot is that Dice Coefficient outliers do not necessarily translate to Hausdorff Distance outliers, as only three of the Dice outlier samples were also Hausdorff outliers. As a result, we can confirm that whilst the proportion of values for both metrics is maintained, the outlier sample distribution is quite different.

The sensitivity values of the model are fairly close to the Dice Coefficient values for each class. The specificity is more complicated to draw correlations with, as the values are extremely high, with a very small standard deviation. An expected correlation is that higher sensitivity results in a lower specificity value for the particular class. We once again refer to the box plots for both the sensitivity and specificity to analyse each metric more closely, shown in Figure 11.

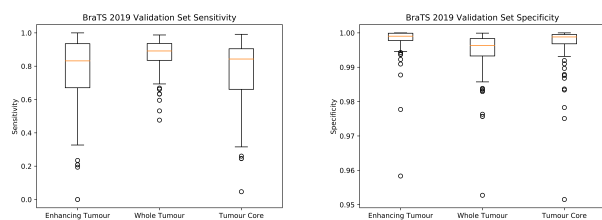


FIGURE 11. Box and whisker plots for the 2019 BraTS validation set Sensitivity and Specificity.

The first observation made is that the the specificity plots' interquartile ranges are near opposite of sensitivity. It is also noteworthy however, that most of the specificity scores are between 0.99 and 1. High specificity values show that the model is very good at avoiding false positives. This aids in the assumption that the model would be capable of avoiding erroneous classification across classes. Another important question to ask with regard to false positives outside of the target classes is how the proposed system behaves when classifying healthy brains. One may make the assumption that false positives are handled well due to the high specificity. This is more difficult to assess, since the BraTS dataset does not contain any full MRI sequences of healthy brains.

A more visual representation of the results is shown in Figure 12. Both samples in the image were taken from a

holdout sample of the BraTS 2019 training dataset (unseen during model training) to showcase the model's predictions against the expert ground truth segmentations. The image shows the segmentation of an LGG and HGG sample from the holdout set.

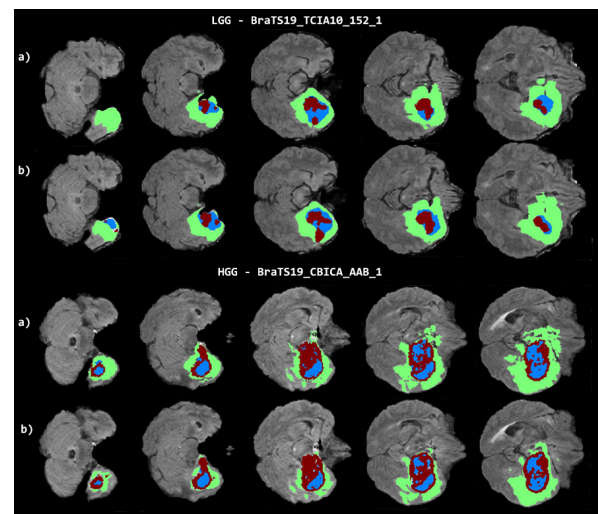


FIGURE 12. a) Ground truths from the BraTS 2019 training dataset vs. (b) the proposed model's predictions of unseen LGG and HGG patients. Left-to-right transitions are displayed at intervals of 5 slices.

An initial observation from the image is that the model performed the HGG segmentation more accurately than for the LGG sample. This may be a result of the data imbalance in the dataset. The most notably distinct slice is the first image from the LGG sample, where the model falsely predicted the edema as a multi-class segment. The other slices are fairly well classified in line with the BraTS ground truths. Another set of comparisons is shown in Figure 13.

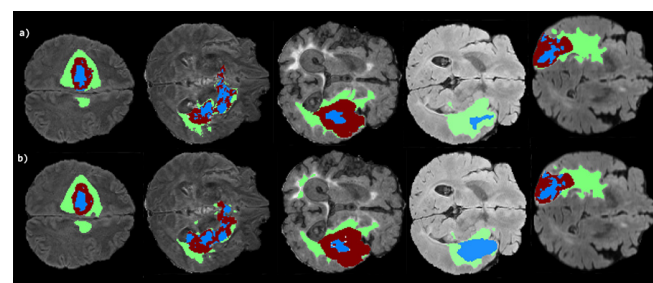


FIGURE 13. a) Ground truths from the BraTS 2019 training dataset vs. (b) the proposed model's predictions of 5 unseen patient cases. Slices were taken from roughly the middle index of each output volume.

Figure 13 compares the model's performance on five separate patients against expert ground truths. The second and fifth sample were selected specifically as they are interesting cases. In the second scan, one may observe how the tumor structure is quite complex. This may have caused the model to overestimate the enhancing tumor regions in its predictions, although the overall shape of the pathology was maintained. The fourth sample was an LGG patient with no enhancing tumor segment. Whilst the model predicted

this correctly, the non-enhancing tumor was overestimated compared to the ground truth. Otherwise, the rest of the samples were fairly well predicted by the model.

C. INTERNAL EVALUATION

As discussed in Section IV-D, two other models were built internally to assess the proposed approach: a standard U-Net and a residual U-Net variant. These models used the same data, training split, and hyperparameters as the final model. The results of the model comparison on the BraTS 2019 validation data are presented in Tables XIX and XX. Since the post-processing experiments were performed on the U-Net++ model, evaluation of the results without post-processing are also tabulated to avoid any form of bias towards the proposed model.

TABLE XIX. Dice Coefficient and Hausdorff Distance for the internal models, with and without constant zero threshold on the BraTS 2019 Validation Dataset. Best scores in bold.

Configuration	Dice Coefficient			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC
Standard U-Net	0.6279	0.8737	0.7725	6.7620	8.4077	9.4486
Residual U-Net	0.6428	0.8433	0.7540	9.0067	11.6163	11.9439
Proposed U-Net++	0.6920	0.8709	0.7824	6.8001	8.3279	9.50
Standard U-Net (Constant Thresh.)	0.6502	0.8737	0.7724	5.9730	8.4096	9.4494
Residual U-Net (Constant Thresh.)	0.6720	0.8433	0.7537	8.3355	11.6119	11.9473
Proposed U-Net++ (Constant Thresh.)	0.7192	0.8712	0.7817	4.6863	8.2157	9.4752

TABLE XX. Sensitivity and specificity for the internal models, with and without constant zero threshold on the BraTS 2019 Validation Dataset. Best scores in bold.

Configuration	Sensitivity			Specificity		
	ET	WT	TC	ET	WT	TC
Standard U-Net	0.7684	0.8859	0.7948	0.9962	0.9934	0.9952
Residual U-Net	0.6840	0.8664	0.7625	0.9981	0.9915	0.9956
Proposed U-Net++	0.7208	0.8654	0.7655	0.9979	0.9945	0.9969
Standard U-Net (Constant Thresh.)	0.7708	0.8859	0.7948	0.9962	0.9934	0.9952
Residual U-Net (Constant Thresh.)	0.6924	0.8663	0.7621	0.9981	0.9915	0.9956
Proposed U-Net++ (Constant Thresh.)	0.7232	0.8671	0.7631	0.9980	0.9944	0.9969

The main priorities for the selection of the proposed U-Net++ architecture from the tests in Section V-A were the Dice Coefficient and Hausdorff Distance. This is also shown in the internal evaluation, where the proposed model outperformed both the standard U-Net and the residual model. Interestingly, the standard U-Net model obtained the highest WT Dice score and also the highest TC Hausdorff Distance score by a very small margin. This may simply mean that whilst the standard U-Net struggled to predict the ET segments correctly compared to the proposed model, it was slightly better at identifying the edema sections.

The sensitivity scores obtained by the standard U-Net model were nonetheless the highest out of all three approaches assessed in the internal evaluation. This implies

that whilst the standard U-Net's predicted tumor sections were not nearly as accurate as the proposed model, it was still better at avoiding classifying false negatives in the MR images. This raises an interesting possibility for future work, as combining both models in an ensemble-like architecture may result in an improvement over the proposed model's sensitivity score.

D. EXTERNAL EVALUATION

The model presented in this paper was also evaluated against peer reviewed work with published results using the BraTS 2019 validation data. The selection of approaches in this table was mostly based on BraTS 2019 publicly available papers, whilst maintaining diversity in the selected approaches. The comparison of all the results are presented in Tables XXI and XXII.

TABLE XXI. Comparison between the Dice Coefficient and Hausdorff Distance of the proposed approach and some state-of-the-art approaches on the BraTS 2019 Validation Dataset. Values of '-' refer to unreported data. Best scores in bold.

Method	Dice Coefficient			Hausdorff Distance		
	ET	WT	TC	ET	WT	TC
Amian and Soltaninejad [47]	0.71	0.84	0.74	10.11	14.00	16.06
Wang <i>et al.</i> [48]	0.737	0.894	0.807	5.994	5.677	7.357
Murugesan <i>et al.</i> [45]	0.784	0.897	0.780	-	-	-
Hamgahalam <i>et al.</i> [49]	0.767	0.897	0.790	4.600	6.900	8.400
Myronenko <i>et al.</i> [46]	0.800	0.894	0.834	3.921	5.890	6.562
Ours	0.719	0.871	0.782	4.686	8.216	9.475

TABLE XXII. Comparison between the Sensitivity and Specificity of the proposed approach and some state-of-the-art approaches on the BraTS 2019 Validation Dataset. Best scores in bold. Some entries removed due to unreported sensitivity and specificity.

Method	Sensitivity			Specificity		
	ET	WT	TC	ET	WT	TC
Amian and Soltaninejad [47]	0.68	0.82	0.74	1.00	0.99	1.00
Wang <i>et al.</i> [48]	0.766	0.897	0.826	0.998	0.995	0.996
Hamgahalam <i>et al.</i> [49]	0.769	0.913	0.777	0.999	0.994	0.998
Ours	0.723	0.867	0.763	0.998	0.994	0.997

The first of the tabulated external approaches by Amian and Soltaninejad [47] makes use of a two-way pipeline. One pathway consists of a standard U-Net which takes the full resolution images as input, whilst the other uses a residual model similar to the approach by [19] on lower resolution samples. This approach was surpassed by the proposed model on all metrics barring the specificity, although this may be due to the rounding used by the authors.

The approach by Wang *et al.* [48] is the first of the remaining tabulated techniques which surpassed the proposed approach. The pipeline in [48] is similar to the model explored in this paper, making use of a standard U-Net. The main difference explored by [48] is the use of a smart patching strategy with the patch windows being generated depending on an offset from the brain boundaries. The patching strategy results in two separate patching cycles which are composed of brain voxels from the MR images.

The next study by Murugesan *et al.* [45] uses a more complex pipeline of multiresolution and multidimensional models. These networks are made up of variations of Inception Networks, Residual Inception Networks, and Dense Networks. Each of the three BraTS tumor classes was segmented using a separate ensemble of these networks, combining the networks' output using element-wise addition operations. The work by [45] also made use of a post-processing approach which removed small clusters of predicted voxels. This approach is similar in theory to the post-processing applied on our final model using constant ET voxel thresholding.

Another ensemble method was explored by Hamghalam *et al.* [49] who made use of a Generative Adversarial Network (GAN) to generate synthetic images from the BraTS data. These 'fake' input samples were used in combination with the data using the FLAIR, T_{1ce} , and T_{2w} sequence types. Three separate, fully connected networks were used to cater for each of the axial, coronal, and sagittal planes in the MR images. Another factor of note in [49] is how the authors omitted the T_{1w} BraTS samples from the experiment.

The final work shown in Tables XXI and XXII is the approach by Myronenko *et al.* [46]. The authors made use of a model which was very similar to the submission in [40] which finished first place in BraTS 2018, as discussed in Section III-B. As per the 2018 submission, the input patch size for this experiment was once again very large, using dimensions of $160 \times 192 \times 128$. The 2020 submission also follows an encoder-decoder model approach, and obtained very high scores across all of the reported metrics, much like the 2018 model.

VI. CONCLUSIONS

A. RESEARCH OUTCOMES & LIMITATIONS

Reviewing the established aim and objectives of this paper, the results show that the model performs segmentation of the multiclass brain tumor segments automatically without any human intervention. Moreover, we have adapted the U-Net++ model in a unique way with a number of experiments which showcase the effects of modifications applied to the architecture and the extent of their improvements or otherwise. An example would be how the ablation study for data augmentation showed a significant improvement across all of the model metrics, and how reducing the number of convolutional blocks came with minimal disadvantages.

One should note that given the size of the models, the training time is substantial. This led to one of the main limitations of the project where the models had to be trained on Cloud instances for the increased virtual memory. This also led to personal costs as no funds were allocated for Cloud services. Nonetheless, this was mitigated slightly owing to the reduced model parameters from using only half of the convolutional blocks. Moreover, the earlier models which leveraged upsampled features from the encoder directly were runnable on local hardware, which also assisted in this regard.

B. FUTURE WORK

There are a number of opportunities to explore when attempting to improve the model's predictions. Starting with the pre-processing pipeline, one possibility would be to swap out the current cropping process with the smart patching strategy leveraged by [48]. The current method follows the aforementioned online repository³. The two-phase patching strategy used in [48] could contribute to higher quality input samples for the model, also reducing the possibility of the current cropping method erroneously removing brain voxels from the input MR images. An additional test which could be performed is the inclusion of FLAIR samples in the bias-field correction step of the pre-processing pipeline and checking if this contributes positively to model training.

Moreover, additional pre-processing steps could be followed, such as using the intensity landmark normalization technique by [34], in conjunction with, or instead of the z-score normalization. The use-case for this technique is to address intensity inhomogeneities from separate medical institutions and devices. The z-score normalization itself may also be adjusted, as it is currently performed on the whole dataset, rather than on a per-patient basis. Performing the z-score normalization at an individual case level as in [50] could improve the quality of images' intensity distribution.

There are also possible improvements on the post-processing side. In the current pipeline, post-processing is applied to correct predictions where scans with no ET segment falsely have some voxels classified as ET by the model. [51] explored an additional post-processing technique to tackle the opposite scenario where scans contained an ET segment but this was not predicted by the model. Following the observations in [51], it is possible that the model wrongly labelled ET voxels as peritumoral edema (label 2) in these predictions. An intensity-clustering technique is leveraged by [51] to identify and correct these cases. This would boost the ET segmentation scores of the proposed model substantially, as these cases have an ET score of 0 (out of 1), which reduces the average score on the 125 BraTS validation data samples substantially.

Moving on to the model itself, ensembling is one possible approach which could provide improved results. In this case, we may refer to two different variations of ensembles. Starting with the internal models discussed in Section V-C, one could attempt to ensemble the standard U-Net model with the proposed U-Net++ adaptation, attempting to reap the benefits of both the higher sensitivity and Dice scores achieved by each model respectively. There is also the possibility to train each model either across separate folds (such as five-fold validation), or using multiple separate training runs. Combining the output of each of these models could result in improved segmentation performance.

We may go even further using the proposed model, such as having dedicated networks/paths as in [31] for the HGG and LGG samples. This latter approach could nonetheless

³<https://github.com/ellisdg/3DUnetCNN>

be inconclusive given the imbalance between the HGG and LGG images in the BraTS data. The work by [49] discussed in Section V-D also proposed an interesting approach, using a separate network for each of the axial, coronal, and sagittal dimensions of the MR images.

Other improvements could be applied to the proposed model as-is, such as further testing using separate dropout values. Given the training time constraints described in Section VI-A, testing was only performed using the dropout value of 0.3. Exploring other values of dropout with the proposed model could result in reducing overfitting of the model even further, combined with the data augmentation pipeline already in place.

C. FINAL REMARKS

In this paper, we presented an automatic model for brain tumor segmentation with positive results obtained on the BraTS 2019 validation data. The modifications applied to the model architecture make it more compact, being half the size of the original U-Net++ model [12]. This project provides useful contributions to the field owing to the results obtained from the documented experiments. Earlier versions of this model were presented in peer-reviewed conferences [43], [44], that lead to more extensive research that we presented in this paper. This is especially true since these variants of the proposed model were evaluated on an unseen holdout sample of the BraTS 2019 training dataset.

The benefits of the proposed model stem from the complex, yet low-parameter architecture inherent to U-Net and U-Net++, now packaged in a smaller, more accessible model. The aforementioned experiments provide insight to other researchers, explaining how adjusting certain model features or improving input and output image quality using pre-processing and post-processing increased the final scores of the model. The experiments also highlight which modifications resulted in improvements in the Dice Coefficient, and those which favored sensitivity instead.

VII. SUPPLEMENTARY MATERIAL

A. POST-PROCESSING ANALYSIS

leading to the identification of two groups: : patients with no ET voxels in the ground truth but with ET voxels in the model prediction, and the opposing scenario, where patients with ET voxels in the ground truths had none predicted by the model. Since the miniscule amount of ET voxels generated by the model was suspected to follow a detectable pattern, it was decided to pursue the first set of these two patient groups. To verify this hypothesis, an equivalent experiment was performed on the unseen holdout samples from the BraTS 2019 Training Dataset, as shown in Table XXIII. The tabulated samples show that the same pattern is also visible on the training dataset, with an example shown in Figure 14.

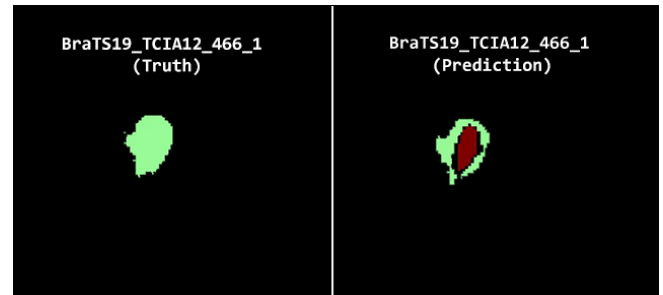


FIGURE 14. Left - Ground truth, Right - Prediction. An example of a falsely predicted ET segment in a sample from the BraTS 2019 Training Dataset.

TABLE XXIII. BraTS 2019 Training Dataset cases with ET Dice score of 0. The tabulated records from the result file show that the same pattern exists in the training data, where the model predicts ET voxels in scans with no actual ET segment in the original image.

Label	Dice Coefficient			NET	Voxels	
	ET	WT	TC		ET	Edema
BraTS19_2013_29_1	0	0.9339	0.7173	10,938	24	41,162
BraTS19_2013_9_1	0	0.8964	0.7669	9,596	10	4,975
BraTS19_TCIA12_466_1	0	0.9105	0.5029	9,681	567	82,176
BraTS19_TCIA13_630_1	0	0.9145	0.7322	25,012	182	71,083

Following the above observations, two possible approaches were explored when attempting to identify patterns in how the model wrongly generates ET segments: a) searching for an ET/NET voxel ratio as a threshold and setting the values below to 0, or b) using a constant amount of voxels as a threshold and setting the values below to 0. To check the voxel ratios scenario, comparisons for the ET (label 4) predictions were compared against the predictions for label 1 and label 2, shown in Table XXIV. The equivalent, but constant voxel amounts were already obtainable from Table XIII.

TABLE XXIV. Ratios of occurrences for label 4 vs. labels 1 and 2 on the BraTS 2019 Validation Dataset for the samples with wrongly predicted ET segments.

Label	Dice Coefficient			Voxels ET/NET	Voxels ET/(NET + Edema)
	ET	WT	TC		
BraTS19_TCIA09_248_1	0	0.9210	0.5672	0.0051	0.0014
BraTS19_TCIA10_127_1	0	0.8941	0.8713	0.0018	0.0012
BraTS19_TCIA10_195_1	0	0.9476	0.7854	0.0300	0.0114
BraTS19_TCIA10_232_1	0	0.8965	0.6610	0.0025	0.0012
BraTS19_TCIA10_609_1	0	0.9514	0.9200	0.0001	0.0001
BraTS19_TCIA10_614_1	0	0.9301	0.3670	0.0174	0.0028

From Table XXIV, a ratio threshold of 0.03 for the ET/NET ratio precisely covers all of the tabulated occurrences. A threshold of 0.04 was nonetheless selected to pad out the selection slightly. Before proceeding with the zero thresholding, it was also imperative to confirm whether such a threshold would be falsely removing any correctly predicted ET segments with small ratios. Thus, all cases where the label 4 ET/NET ratio was less than 0.04 were examined, including samples with an ET score greater than 0 in the result file. One can notice a number of samples which would potentially be erroneously post-processed, shown in Table XXV.

TABLE XXV. BraTS 2019 Validation Dataset cases which would have ET segments erroneously set to 0 with a voxel ratio threshold of 0.04 ET/NET.

Label	Dice Coefficient			Voxels ET/NET
	ET	WT	TC	
BraTS19_TCIA10_220_1	0.1075	0.9302	0.8091	0.0019
BraTS19_TCIA10_239_1	0.5526	0.9174	0.7849	0.0126
BraTS19_TCIA10_647_1	0.3333	0.9240	0.6086	0.0010
BraTS19_TCIA12_339_1	0.0076	0.8549	0.0936	0.0044
BraTS19_TCIA13_611_1	0.0215	0.7591	0.2607	0.0053
BraTS19_TCIA13_616_1	0.6669	0.9184	0.8399	0.0135
BraTS19_TCIA13_617_1	0.1374	0.8096	0.7085	0.0020
BraTS19_TCIA13_638_1	0.3217	0.8565	0.6007	0.0307
BraTS19_TCIA13_643_1	0.0061	0.8371	0.6632	0.0003

Table XXV shows that thresholding by voxel ratio does lead to quite the number of samples being wrongly modified and transformed into false negatives. The same test was thus performed to identify the wrongly processed samples in the case of the constant threshold of up to 200 ET voxels, as shown in Table XXVI.

TABLE XXVI. BraTS 2019 Validation Dataset cases which would have ET segments erroneously set to 0 with a constant threshold of 200 ET voxels.

Label	Dice Coefficient			ET Voxels
	ET	WT	TC	
BraTS19_TCIA10_220_1	0.1075	0.9302	0.8091	65
BraTS19_TCIA10_647_1	0.3333	0.9240	0.6086	6
BraTS19_TCIA12_339_1	0.0076	0.8549	0.0936	8
BraTS19_TCIA13_611_1	0.0215	0.7591	0.2607	148
BraTS19_TCIA13_617_1	0.1374	0.8096	0.7085	74
BraTS19_TCIA13_643_1	0.0061	0.8371	0.6632	31

The results show that the constant thresholding approach is less damaging overall and thus is the best approach out of the two discussed.

REFERENCES

- [1] V. Anitha and S. Murugavalli, "Brain tumour classification using two-tier classifier with adaptive segmentation technique," *IET computer vision*, vol. 10, no. 1, pp. 9–17, 2016.
- [2] K. D. Miller, M. Fidler-Benaoudia, T. H. Keegan, H. S. Hipp, A. Jemal, and R. L. Siegel, "Cancer statistics for adolescents and young adults, 2020," *CA: A Cancer Journal for Clinicians*, 2020.
- [3] H. Sung, J. Ferlay, R. L. Siegel, M. Laversanne, I. Soerjomataram, A. Jemal, and F. Bray, "Global cancer statistics 2020: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries," *CA: a cancer journal for clinicians*, 2021.
- [4] N. Gordillo, E. Montseny, and P. Sobrevilla, "State of the art survey on mri brain tumor segmentation," *Magnetic resonance imaging*, vol. 31, no. 8, pp. 1426–1438, 2013.
- [5] G. P. Mazzara, R. P. Velthuisen, J. L. Pearlman, H. M. Greenberg, and H. Wagner, "Brain tumor target volume determination for radiation treatment planning through automated mri segmentation," *International Journal of Radiation Oncology* Biology* Physics*, vol. 59, no. 1, pp. 300–312, 2004.
- [6] J. Liu, M. Li, J. Wang, F. Wu, T. Liu, and Y. Pan, "A survey of mri-based brain tumor segmentation methods," *Tsinghua Science and Technology*, vol. 19, no. 6, pp. 578–595, 2014.
- [7] B. H. Menze, A. Jakab, S. Bauer, J. Kalpathy-Cramer, K. Farahani, J. Kirby, Y. Burren, N. Porz, J. Slotboom, R. Wiest, et al., "The multimodal brain tumor image segmentation benchmark (brats)," *IEEE transactions on medical imaging*, vol. 34, no. 10, pp. 1993–2024, 2014.
- [8] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos, "Advancing the cancer genome atlas glioma mri collections with expert segmentation labels and radiomic features," *Scientific Data*, vol. 4, Sept. 2017.
- [9] S. Bakas, M. Reyes, A. Jakab, S. Bauer, M. Rempfler, A. Crimi, R. T. Shinohara, C. Berger, S. M. Ha, M. Rozycki, et al., "Identifying the best machine learning algorithms for brain tumor segmentation, progression assessment, and overall survival prediction in the brats challenge," *arXiv preprint arXiv:1811.02629*, 2018.
- [10] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos, "Segmentation labels and radiomic features for the pre-operative scans of the tcga-1gg collection," July 2017.
- [11] S. Bakas, H. Akbari, A. Sotiras, M. Bilello, M. Rozycki, J. Kirby, J. Freymann, K. Farahani, and C. Davatzikos, "Segmentation labels and radiomic features for the pre-operative scans of the tcga-gbm collection," July 2017.
- [12] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pp. 3–11, Springer, 2018.
- [13] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.
- [14] M. Ghaffari, A. Sowmya, and R. Oliver, "Automated brain tumour segmentation using multimodal brain scans, a survey based on models submitted to the brats 2012-18 challenges," *IEEE Reviews in Biomedical Engineering*, pp. 1–1, 2019.
- [15] D. W. McRobbie, E. A. Moore, M. J. Graves, and M. R. Prince, *Mri from picture to proton*. Cambridge university press, 2017.
- [16] B. Kayalibay, G. Jensen, and P. van der Smagt, "Cnn-based segmentation of medical imaging data," *arXiv preprint arXiv:1701.03056*, 2017.
- [17] H. Chen, Q. Dou, L. Yu, and P.-A. Heng, "Voxresnet: Deep voxelwise residual networks for volumetric brain segmentation," *arXiv preprint arXiv:1608.05895*, 2016.
- [18] Q. Dou, H. Chen, Y. Jin, L. Yu, J. Qin, and P.-A. Heng, "3d deeply supervised network for automatic liver segmentation from ct volumes," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 149–157, Springer, 2016.
- [19] F. Isensee, P. Kickingereder, W. Wick, M. Bendszus, and K. H. Maier-Hein, "Brain tumor segmentation and radiomics survival prediction: contribution to the brats 2017 challenge," in *International MICCAI Brainlesion Workshop*, pp. 287–297, Springer, 2017.
- [20] F. Chen, Y. Ding, Z. Wu, D. Wu, and J. Wen, "An improved framework called du++ applied to brain tumor segmentation," in *2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 85–88, IEEE, 2018.
- [21] Z. Wu, F. Chen, and D. Wu, "A novel framework called hdu for segmentation of brain tumor," in *2018 15th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, pp. 81–84, IEEE, 2018.
- [22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.
- [23] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on mri," *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 102–127, 2019.
- [24] M.-N. Wu, C.-C. Lin, and C.-C. Chang, "Brain tumor detection using color-based k-means clustering segmentation," in *Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP 2007)*, vol. 2, pp. 245–250, IEEE, 2007.
- [25] T. Chaira and S. Anand, "A novel intuitionistic fuzzy approach for tumour/hemorrhage detection in medical images," *Journal of Scientific and Industrial Research*, 2011.
- [26] J. Juan-Albarracín, E. Fuster-Garcia, J. V. Manjón, M. Robles, F. Aparici, L. Martí-Bonmatí, and J. M. García-Gómez, "Automated glioblastoma segmentation based on a multiparametric structured unsupervised classification," *Plos One*, vol. 10, pp. 1–20, May 2015.
- [27] J. Zhou, K. L. Chan, V. F. H. Chong, and S. M. Krishnan, "Extraction of brain tumor from mr images using one-class support vector machine," in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pp. 6411–6414, IEEE, 2006.
- [28] P. Sriramakrishnan, T. Kalaiselvi, and R. Rajeswaran, "Modified local ternary patterns technique for brain tumour segmentation and volume

- estimation from mri multi-sequence scans with gpu cuda machine,” *Biocybernetics and Biomedical Engineering*, vol. 39, no. 2, pp. 470–487, 2019.
- [29] N. J. Tustison, K. L. Shrinidhi, M. Wintermark, C. R. Durst, B. M. Kandel, J. C. Gee, M. C. Grossman, and B. B. Avants, “Optimal symmetric multimodal templates and concatenated random forests for supervised brain tumor segmentation (simplified) with ants,” *Neuroinformatics*, vol. 13, no. 2, pp. 209–225, 2015.
- [30] M. Soltaninejad, G. Yang, T. Lambrou, N. Allinson, T. L. Jones, T. R. Barrick, F. A. Howe, and X. Ye, “Supervised learning based multimodal mri brain tumour segmentation using texture features from supervoxels,” *Computer methods and programs in biomedicine*, vol. 157, pp. 69–84, 2018.
- [31] S. Pereira, A. Pinto, V. Alves, and C. A. Silva, “Brain tumor segmentation using convolutional neural networks in mri images,” *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1240–1251, 2016.
- [32] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [33] N. J. Tustison, B. B. Avants, P. A. Cook, Y. Zheng, A. Egan, P. A. Yushkevich, and J. C. Gee, “N4itk: improved n3 bias correction,” *IEEE transactions on medical imaging*, vol. 29, no. 6, p. 1310, 2010.
- [34] L. G. Nyúl, J. K. Udupa, and X. Zhang, “New variants of a method of mri scale standardization,” *IEEE transactions on medical imaging*, vol. 19, no. 2, pp. 143–150, 2000.
- [35] K. Kamnitsas, E. Ferrante, S. Parisot, C. Ledig, A. V. Nori, A. Criminisi, D. Rueckert, and B. Glocker, “Deepmedic for brain tumor segmentation,” in *International workshop on Brainlesion: Glioma, multiple sclerosis, stroke and traumatic brain injuries*, pp. 138–149, Springer, 2016.
- [36] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, “Brain tumor segmentation with deep neural networks,” *Medical image analysis*, vol. 35, pp. 18–31, 2017.
- [37] F. Isensee, P. Kickingereder, W. Wick, M. Bendszus, and K. H. Maier-Hein, “No new-net,” in *International MICCAI Brainlesion Workshop*, pp. 234–244, Springer, 2018.
- [38] R. McKinley, R. Meier, and R. Wiest, “Ensembles of densely-connected cnns with label-uncertainty for brain tumor segmentation,” in *International MICCAI Brainlesion Workshop*, pp. 456–465, Springer, 2018.
- [39] R. McKinley, A. Jungo, R. Wiest, and M. Reyes, “Pooling-free fully convolutional networks with dense skip connections for semantic segmentation, with application to brain tumor segmentation,” in *International MICCAI Brainlesion Workshop*, pp. 169–177, Springer, 2017.
- [40] A. Myronenko, “3d mri brain tumor segmentation using autoencoder regularization,” in *International MICCAI Brainlesion Workshop*, pp. 311–320, Springer, 2018.
- [41] B. B. Avants, N. Tustison, and G. Song, “Advanced normalization tools (ants),” *Insight j*, vol. 2, no. 365, pp. 1–35, 2009.
- [42] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [43] N. Micallef, D. Seychell, and C. Bajada, “A nested u-net approach for brain tumour segmentation,” in *OHBM 2020 Annual Meeting*, (Montreal, Canada), 2020.
- [44] N. Micallef, D. Seychell, and C. Bajada, “A nested u-net approach for brain tumour segmentation,” in *2020 IEEE 20th Mediterranean Electrotechnical Conference (MELECON)*, (Palermo, Italy), June 2020.
- [45] G. K. Murugesan, S. Nalawade, C. G. B. Yogananda, B. Wagner, B. Fei, A. Madhuranthakam, and J. A. Maldjian, “Multidimensional and multi-resolution ensemble networks for brain tumor segmentation,” *bioRxiv*, p. 760124, 2019.
- [46] A. Myronenko and A. Hatamizadeh, “Robust semantic segmentation of brain tumor regions from 3d mrIs,” *arXiv preprint arXiv:2001.02040*, 2020.
- [47] M. Amian and M. Soltaninejad, “Multi-resolution 3d cnn for mri brain tumor segmentation and survival prediction,” *arXiv preprint arXiv:1911.08388*, 2019.
- [48] F. Wang, R. Jiang, L. Zheng, B. Biswal, and C. Meng, “Brain-wise tumor segmentation and patient overall survival prediction,” *arXiv preprint arXiv:1909.12901*, 2019.
- [49] M. Hamghalam, B. Lei, and T. Wang, “Brain tumor synthetic segmentation in 3d multimodal mri scans,” *arXiv preprint arXiv:1909.13640*, 2019.
- [50] Z. Jiang, C. Ding, M. Liu, and D. Tao, “Two-stage cascaded u-net: 1st place solution to brats challenge 2019 segmentation task,” in *International MICCAI Brainlesion Workshop*, pp. 231–241, Springer, 2019.
- [51] C. Zhou, C. Ding, Z. Lu, X. Wang, and D. Tao, “One-pass multi-task convolutional neural networks for efficient brain tumor segmentation,” in

International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 637–645, Springer, 2018.



NEIL MICALLEF (GSM’20) received the B.Sc. IT (Hons.) degree in Artificial Intelligence in 2018 from the University of Malta. He recently successfully completed the M.Sc. IT (Hons.) in Artificial Intelligence at the University of Malta. His research interests include computer vision, with a recent focus on neuroimaging. Neil is currently working as a software and application development manager at Transport Malta, the Authority for Transport in the Maltese Islands.



DYLAN SEYCHELL (S’07-GSM’10-M’19-SM’20) received the B.Sc.IT (Hons.) degree in Computer Science and Artificial Intelligence in 2010 and the M.Sc. degree in Artificial Intelligence in 2011 from the University of Malta, Malta. He is currently reading for a Ph.D degree in computer vision with the Department of Communications and Computer Engineering at the University of Malta.

Between 2011 and 2017 he was a Resident Academic at Saint Martin’s Institute of Higher Education where he served as Head of the Computing Department for five years. In 2017, he joined the Department of Artificial Intelligence at the University of Malta as an Assistant Lecturer. His research interests are visual attention, saliency, image manipulation, machine learning and user experience design. He was awarded a number of international awards for his work such as the Gold Seal for e-Excellence at CeBit in 2011, the first prize by the European Satellite Navigation Competition (Living Labs) in 2010 and runner up in 2017. In 2015, Dylan was selected to lead Malta’s Google Developers Group. He also served as a member of the Malta Neuroscience Network and the Malta National AI Taskforce that was responsible for the development of the national AI strategy. Dylan is involved in startups related to technology applied to heritage and tourism. He serves as a technology advisor and coordinator on a number of high-profile projects.



CLAUDE J. BAJADA received his medical degree from the University of Malta in 2010 followed by an MSc and PhD in Cognitive Neuroscience from UCL and the University of Manchester. He is currently a lecturer in Physiology at the University of Malta. His primary research focus is in neuroimaging. He is interested in using large-scale, open data to answer fundamental questions about organisational principles of the cerebral cortex, and its connectivity. He is also interested in

the translation between basic science and clinical utility.

...